Programmazione I

A.A. 2002-03

Costrutti di base

(Lezione XIV , parte I)

Costrutto di selezione "switch"

Prof. Giovanni Gallo
Dr. Gianluca Cincotti
Dipartimento di Matematica e Informatica
Università di Catania
e-mail : {gallo, cincotti}@dmi.unict.it

Un esempio di "if" annidati

- Vogliamo un programma che prenda in input un intero e produca un messaggio diverso a seconda che si immetta il numero 0, 1 oppure 2.
  - Lo strumento a nostra disposizione (fino ad adesso) è l'istruzione "if … else".

Un esempio di "if" annidati (cont.)

```
// prima di questo frammento abbiamo
// ottenuto l'intero x con un qualunque
// metodo per la gestione dell'input

if (x==0) System.out.println("zero");
else if (x==1) System.out.println("uno");
     else if (x==2) System.out.println("due");
          else System.out.println("non capisco");
```

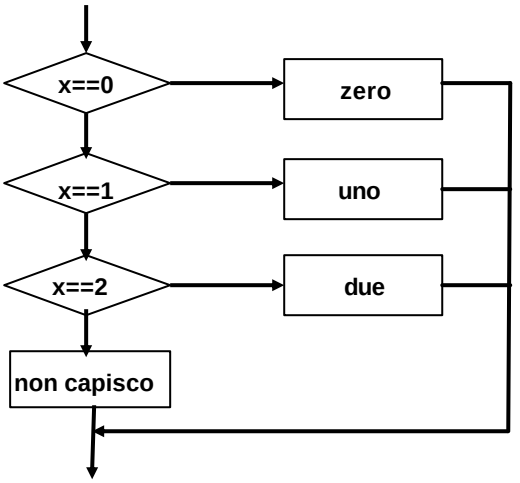
diagramma di flusso

Un esempio di "if" annidati (cont.)

Diagramma di flusso di tre "if" a cascata.

Questa situazione si incontra spesso.

Per renderla più compatta nei programmi invece di una cascata di "if" si può usare lo "switch"



La sintassi dell'istruzione "switch"

```
switch (Espressione)
  {
    case valore_1:
        Istruzione_1;
        break;
    case valore_2 :
        Istruzione_2;
        break;
     …
    default:
        IstruzioneDiDefault;
}
```


Dunque …

```
switch ( x )
   {
     case 0 :
        System.out.println("zero");
        break;
     case 1:
        System.out.println("uno");
        break;
     default:
        System.out.println("non capisco");
  }
```

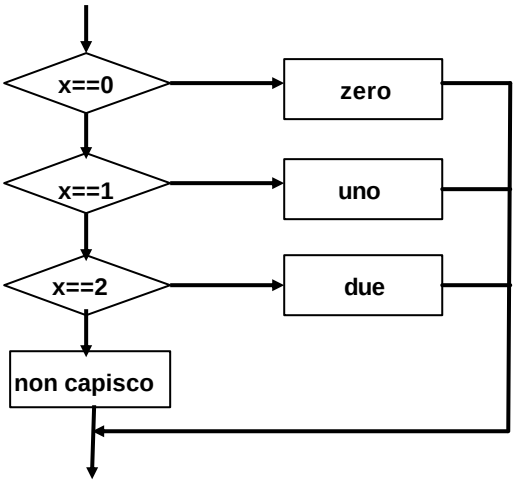
La semantica dell'istruzione "switch"

- L'istruzione *switch* costituisce un'abbreviazione per una sequenza di "if" annidati.
- Essa valuta un'espressione e ne confronta il risultato con i diversi *casi* elencati
  - Ogni *caso* contiene un valore e una lista di istruzioni da eseguire.
- Il flusso di controllo è trasferito alla lista associata con il primo valore uguale all'espressione

… e quindi …

Già visto?

In effetti….



L'istruzione "break"

- L'istruzione *break* pasa il controllo alla fine dell'istruzione *switch.*
- Se non si usa l'istruzione *break,* il flusso di controllo continua ai casi successivi.
  - Qualche volta questo è utile, ma di solito i valori dei diversi casi sono mutuamente esclusivi e al più un caso corrisponde al valore dell'espressione.
    - In generale, non si deve mai dimenticare il comando *break* tra un caso e l'altro (stesso problema nel C !).

L'istruzione "break" (cont.)

```
int x;
   …
switch (x)
   {
    case 0:
         System.out.println("zero");
         break;
    case 1:
         System.out.println("uno");
    default:
         System.out.println("altro");
 }
```

Che succede se l'input x è 1?

Verrà visualizzato:
uno
altro

Ciò dipende dalla assenza del *break* dopo il case 1.

La clausola "default"

- Un'istruzione *switch* può avere un caso di *default.*
  - Il caso di *default* non ha un valore associato ma usa semplicemente la parola riservata `default.`
  - Se è presente il caso di *default*, il controllo è trasferito all'istruzione associata se non ci sono altri casi
  - Se non è presente il caso di *default* e nessun valore corrisponde, il controllo passa all'istruzione successiva all'istruzione *switch*

Limiti di "switch"

- L'espressione di un'istruzione *switch* deve produrre un valore intero o carattere.
  - Non può essere un valore in virgola mobile o una stringa.
- La condizione nell'istruzione *switch* è sempre un'uguaglianza.
  - Non si possono utilizzare altri operatori relazionali.

Limiti di "switch" (cont.)

```
String m;
   …
switch (m)
   {
     case "pippo":
          System.out.println("baudo");
          break;
     case "mike":
          System.out.println("bongiorno");
          break;
     default:
          System.out.println("frizzi?");
}
```

L'istruzione "switch" non può gestire stringhe !!!

Esempi

Es. 01

Es. 02

Programmazione I

A.A. 2002-03

Costrutti di base

(Lezione XIV , parte II)

Costrutto di iterazione "do … while"

Prof. Giovanni Gallo
Dr. Gianluca Cincotti
Dipartimento di Matematica e Informatica
Università di Catania
e-mail : {gallo, cincotti}@dmi.unict.it

Istruzioni iterative

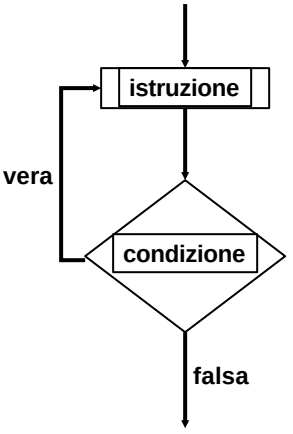
- I costrutti di *iterazione* consentono di eseguire molte volte la stessa istruzione e sono controllati da espressioni booleane.
- In Java esistono *tre* tipi di istruzioni iterative:
  - il ciclo *while*,
  - il ciclo *do…while*,
  - il ciclo *for*.
- Sono equivalenti ma…
  - appropriati in situazioni diverse !

L'istruzione "do … while"

L'istruzione **do-while** a differenza del **while**, controlla il valore della espressione booleana **alla fine** del blocco di istruzioni.
In questo modo quindi il corpo del ciclo verrà eseguito sicuramente almeno una volta.

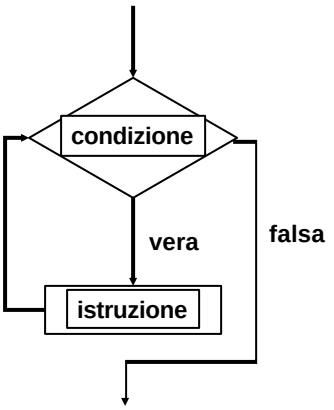
```
do {
        Istruzione;
   } while (espressione_booleana);
```


Semantica del ciclo "do…while"

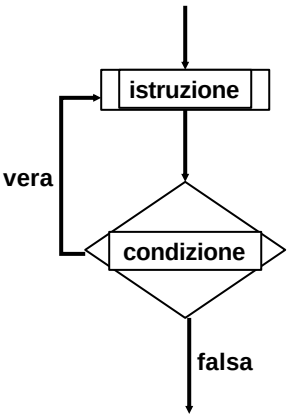


Confronto tra i cicli

ciclo *while*



ciclo *do…while*



Repetita juvant…

- La differenza fra **while** e **do-while** consiste nel fatto che:
  - il corpo del ciclo nel **do-while** viene sempre eseguito almeno una volta (cioè la prima volta);
  - nel **while** invece se la condizione booleana è falsa il corpo del ciclo non viene mai eseguito.

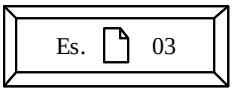
Si può "scappare" dai cicli ?

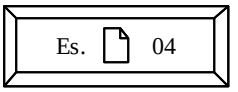
Il comando **break** visto all'interno dell'istruzione "*switch*" può essere usato anche per "scappare" dal body di un ciclo "**while**" o "**do…while**".

Tale pratica non è mai necessaria e rende il codice particolarmente oscuro: **EVITATELO!!!**

Esercitazione

- Lanci ripetuti di un dado …
  - Si lancia un dado finché escono consecutivamente 5 facce uguali;
  - Visualizzare il numero dei lanci che sono stati necessari ed il valore che si è ripetuto.





Fine